

---

**Équipe A\* (208)**

---

**PolyDiff**  
**Document d'architecture logicielle**

**Version 1.10**

# Historique des révisions

Date	Version	Description	Auteur
2024-01-14	1.0	Rédaction de l'introduction	Lahbabi Ryan
2024-01-21	1.1	Rédaction de la partie taille et performance	Lahbabi Ryan
2024-01-22	1.2	Rédaction de la partie cas d'utilisation	Zhiqin Yin
2024-01-22	1.3	Ajout du diagramme de vue de déploiement	Lahbabi Ryan
2024-01-24	1.4	Rédaction du diagramme de vue de processus	Zhiqin Yin
2024-01-24	1.5	Correction du français et ajout d'informations dans objectifs et contraintes architecturaux	Jeremy Rouillard
2024-01-28	1.6	Mettre les diagrammes de classes et de paquetages qui ne changent pas par rapport au projet 2	Zarine Ardekani
2024-01-28	1.7	Révision du contenu du rapport.	Jeremy Rouillard
2024-01-30	1.8	Mettre à jour les diagrammes de la vue logique liés à la création de jeu et au déroulement d'une partie	Zarine Ardekani
2024-01-31	1.9	Diagramme de classe de la reprise vidéo et de la vue d'administration	Zarine Ardekani
2024-01-08	1.10	Modification reliée aux commentaires	Jeremy Rouillard

# Table des matières

<b>1. Introduction</b>	<b>4</b>
<b>2. Objectifs et contraintes architecturaux</b>	<b>5</b>
<b>3. Vue des cas d'utilisation</b>	<b>6</b>
<b>4. Vue logique</b>	<b>10</b>
<b>5. Vue des processus</b>	<b>22</b>
<b>6. Vue de déploiement</b>	<b>30</b>
<b>7. Taille et performance</b>	<b>31</b>

# Document d'architecture logicielle

## 1. Introduction

Le document présente une vue globale de l'architecture logicielle de nos applications web et mobiles. Ce dernier est organisé en six sections principales: les objectifs et contraintes architecturales, la vue des cas d'utilisation, la vue des processus, la vue logique, la vue de déploiement et pour finir la taille et performance de notre architecture logicielle.

Dans la première section, nous présenterons les éléments clés qui influencent l'architecture du logiciel ou du système en développement. La section aborde les objectifs à atteindre ainsi que les contraintes à respecter qui ont un impact significatif sur la conception architecturale. Cette section est importante, car elle établit les fondations sur lesquelles le système sera construit et détermine les limites et les possibilités dans le cadre du développement de nos applications web et mobile.

Dans la deuxième section, nous présentons les cas d'utilisation pertinents du modèle de cas d'utilisation, en mettant l'accent sur les fonctionnalités que nous allons développer dans les prochaines semaines. Les diagrammes de cas d'utilisation permettent de décrire les interactions entre les acteurs et le système pour chaque fonctionnalité.

Dans la troisième section, nous montrerons les parties architecturalement significatives de notre modèle de conception en d'autres termes notre vue logique. Pour illustrer cela, nous utiliserons les diagrammes de paquetages et de classes vu en LOG1410. Vous trouverez les descriptions de chaque paquetage que nous avons jugé pertinentes dans notre modèle de conception architecturale.

Dans la quatrième section, nous présentons les processus significatifs du système sous forme de diagrammes de séquences. Cette section permet de décrire les interactions entre les différents processus du système et les données échangées entre eux.

Dans la cinquième, nous décrirons la vue de déploiement des différentes parties du système. Vous trouverez les nœuds physiques et les interconnexions utilisées, notamment les protocoles de communication utilisés. Cela sera illustré sous la forme d'un diagramme de déploiement.

La dernière section se concentrera sur les caractéristiques de taille et de performance du logiciel. Elle présente une analyse des exigences en termes de dimensions, comme la quantité de stockage ou la mémoire nécessaire, et de capacités de performance telles que la vitesse de traitement. Cette partie est essentielle pour comprendre comment ces aspects influencent l'architecture et le design du logiciel, en établissant des paramètres clés qui doivent être pris en compte durant le processus de développement.

## 2. Objectifs et contraintes architecturaux

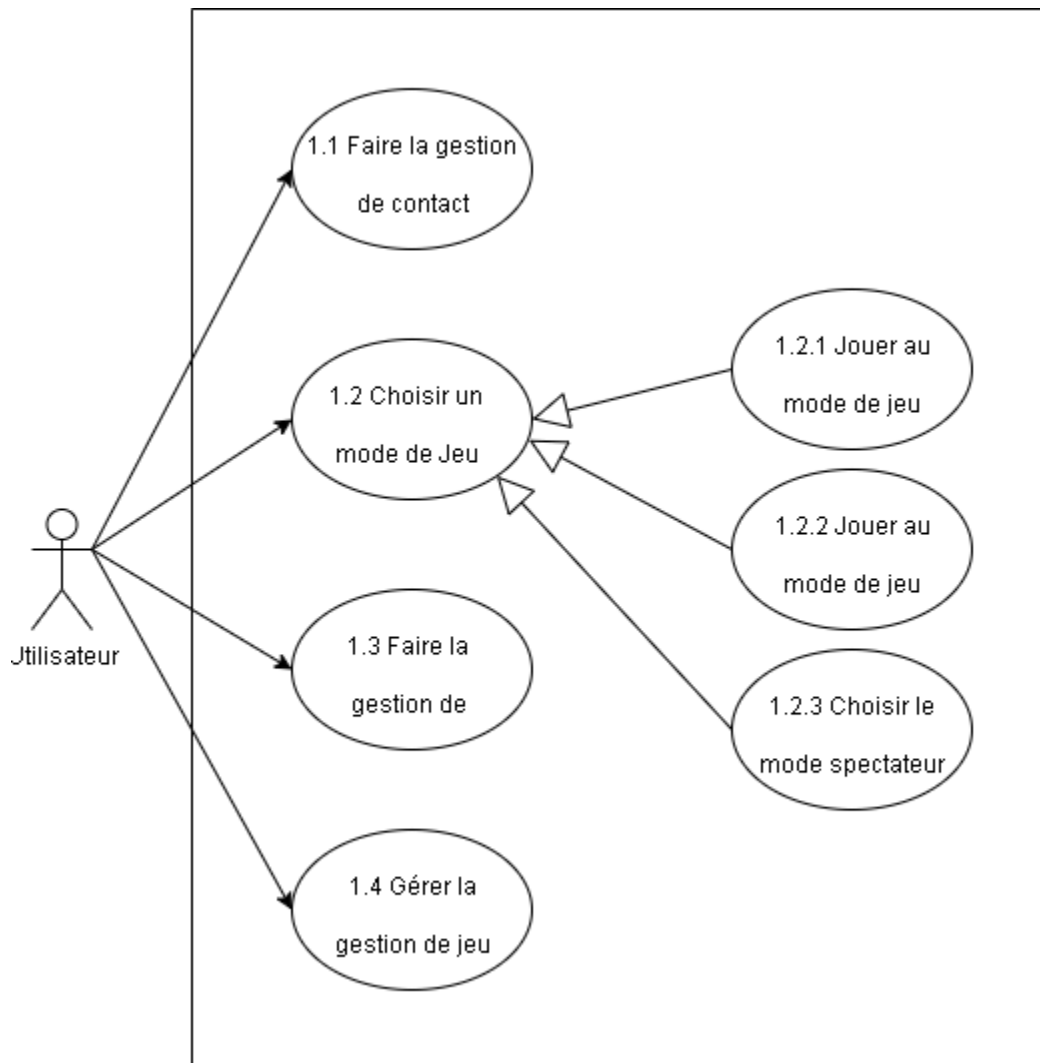
Dans cette section, nous abordons les objectifs et contraintes qui influencent de manière significative l'architecture de notre projet. Notre principal objectif est d'élaborer une architecture suffisamment flexible pour intégrer aisément de nouvelles fonctionnalités. Depuis le lancement du projet de développement web, l'ajout continu de fonctionnalités a exigé un haut niveau de réutilisation du code. Notre ambition est de développer cette application pour mobile sur tablette en utilisant Dart-Flutter. Avec Dart au sein du cadriciel Flutter, nous rencontrons certaines contraintes spécifiques, telles que la gestion efficace de l'état des applications, l'intégration avec un système de base de données et réseaux, ainsi qu'une dépendance envers l'écosystème évolutif des packages de Flutter.

Dans la perspective des développements futurs, l'introduction de nouvelles fonctionnalités nécessitera une réévaluation de certains aspects architecturaux. En particulier, l'intégration de fonctionnalités liées aux comptes utilisateurs nous oblige à repenser sérieusement la sécurité et la confidentialité. Cela implique la mise en place de systèmes d'authentification solides, le chiffrement des données des utilisateurs, et l'établissement de protocoles rigoureux pour la protection des données personnelles, en accord avec des normes réglementaires telles que le RGPD. Il sera également crucial d'assurer une gestion sécurisée des sessions et des jetons d'accès, ainsi qu'une communication sécurisée entre l'application et les serveurs.

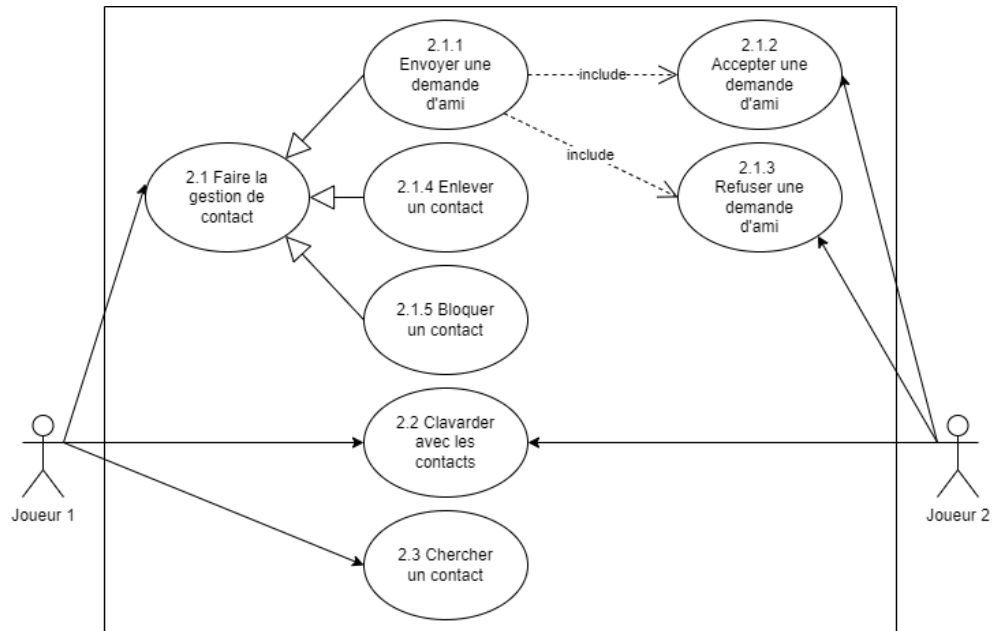
La portabilité représente un autre aspect clé, car l'application doit fonctionner sans heurts sur différents formats de tablettes, tout en offrant une expérience utilisateur uniforme. L'emploi de Flutter avantage cette portabilité.

Concernant les coûts et l'échéancier, l'utilisation de Dart-Flutter promet de réduire le temps de développement grâce à la réutilisation substantielle du code sur diverses plateformes. Néanmoins, il est impératif de prévoir des ressources adéquates pour des tests exhaustifs sur chaque plateforme visée, ainsi que pour des ajustements et des optimisations spécifiques à chaque environnement de plateforme.

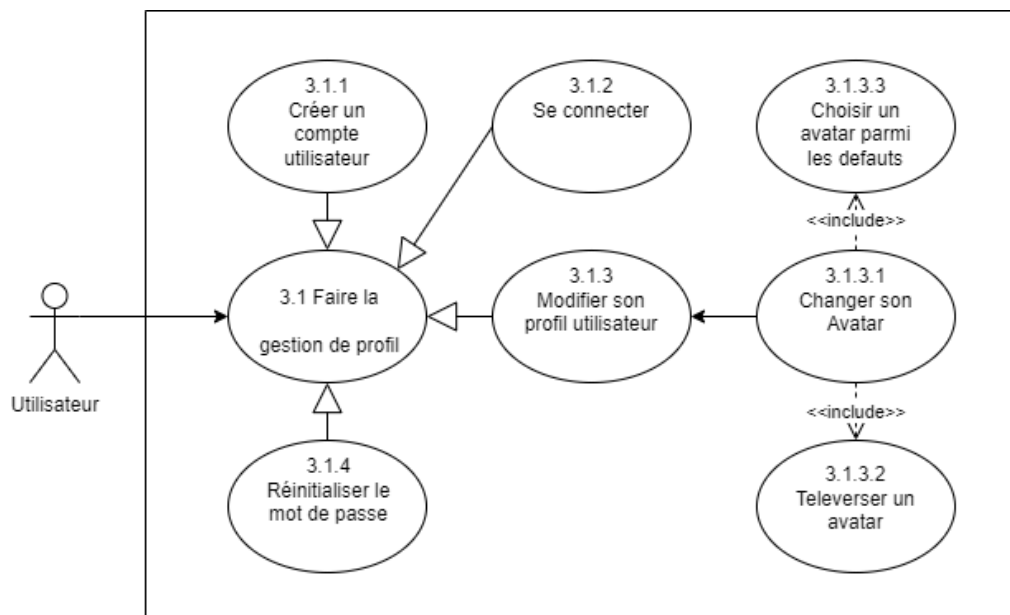
### 3. Vue des cas d'utilisation



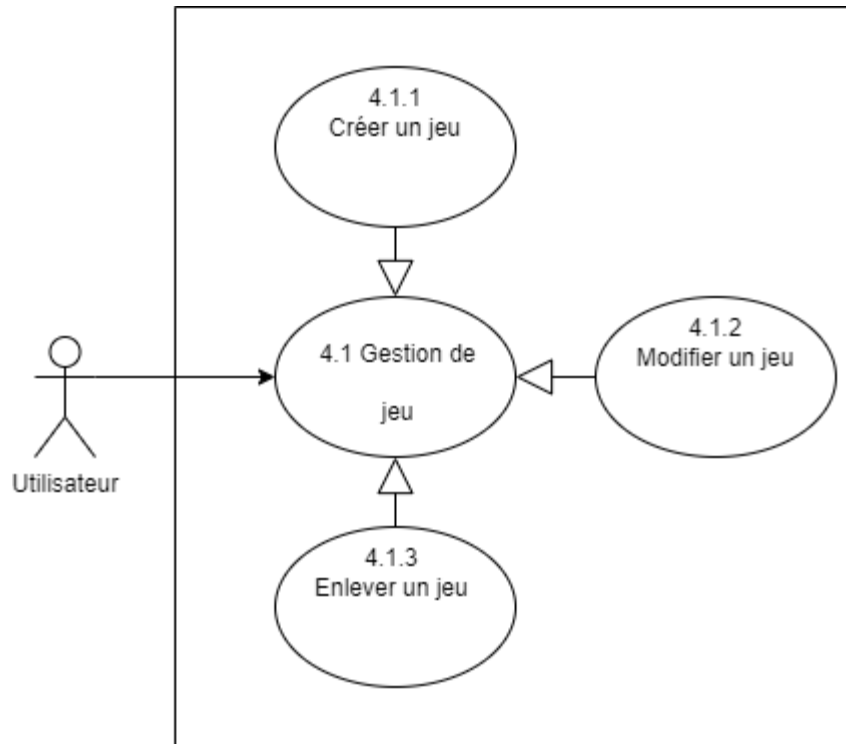
*Figure 3.1: Cas d'utilisation de notre projet vue général.*



**Figure 3.2: Cas d'utilisation du système de contacts.**

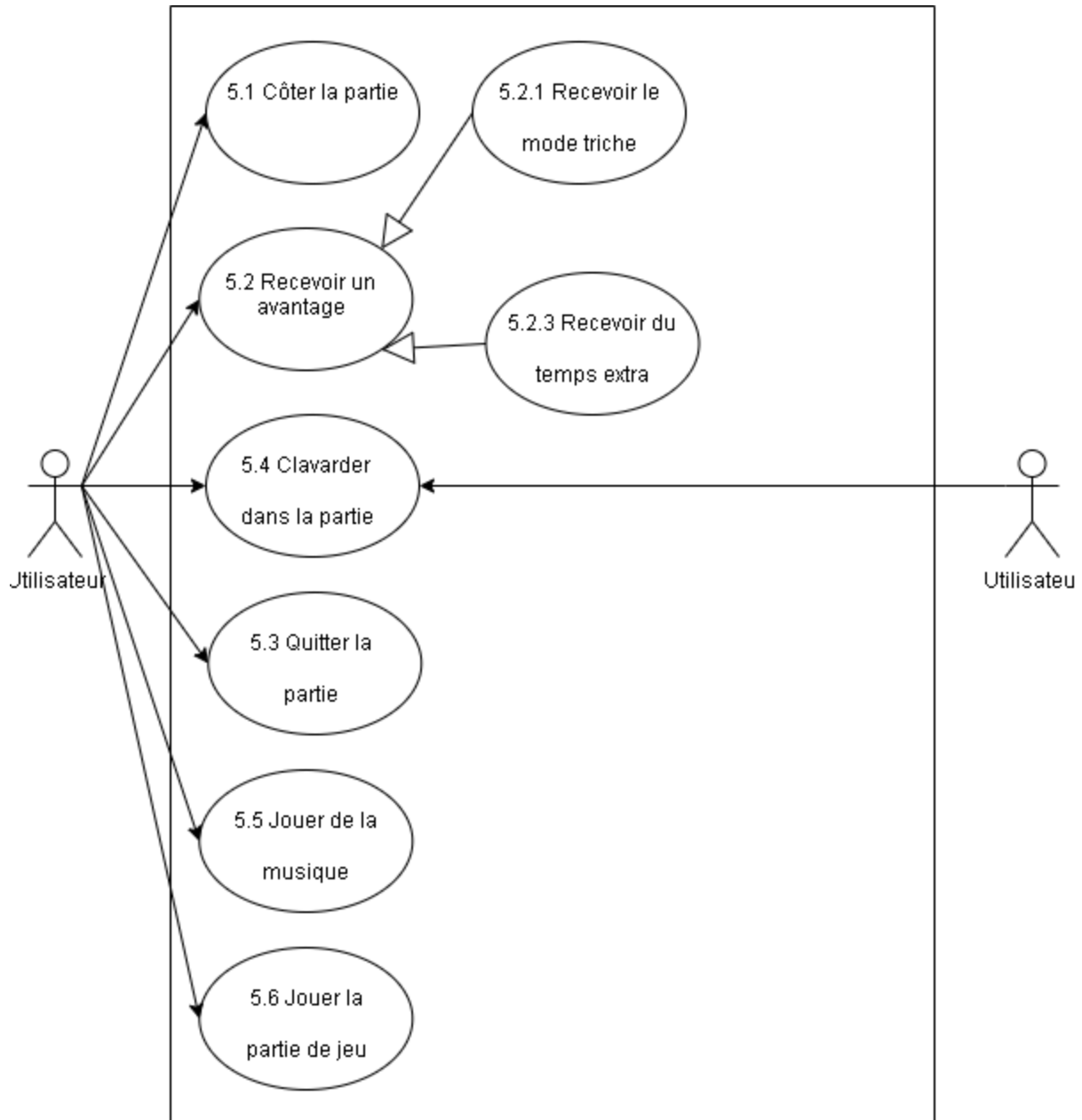


**Figure 3.3: Cas d'utilisation du système de profil.**



*Figure 3.4: Cas d'utilisation du système de paramètre de jeu.*





*Figure 3.5: Cas d'utilisation du système durant un jeu.*

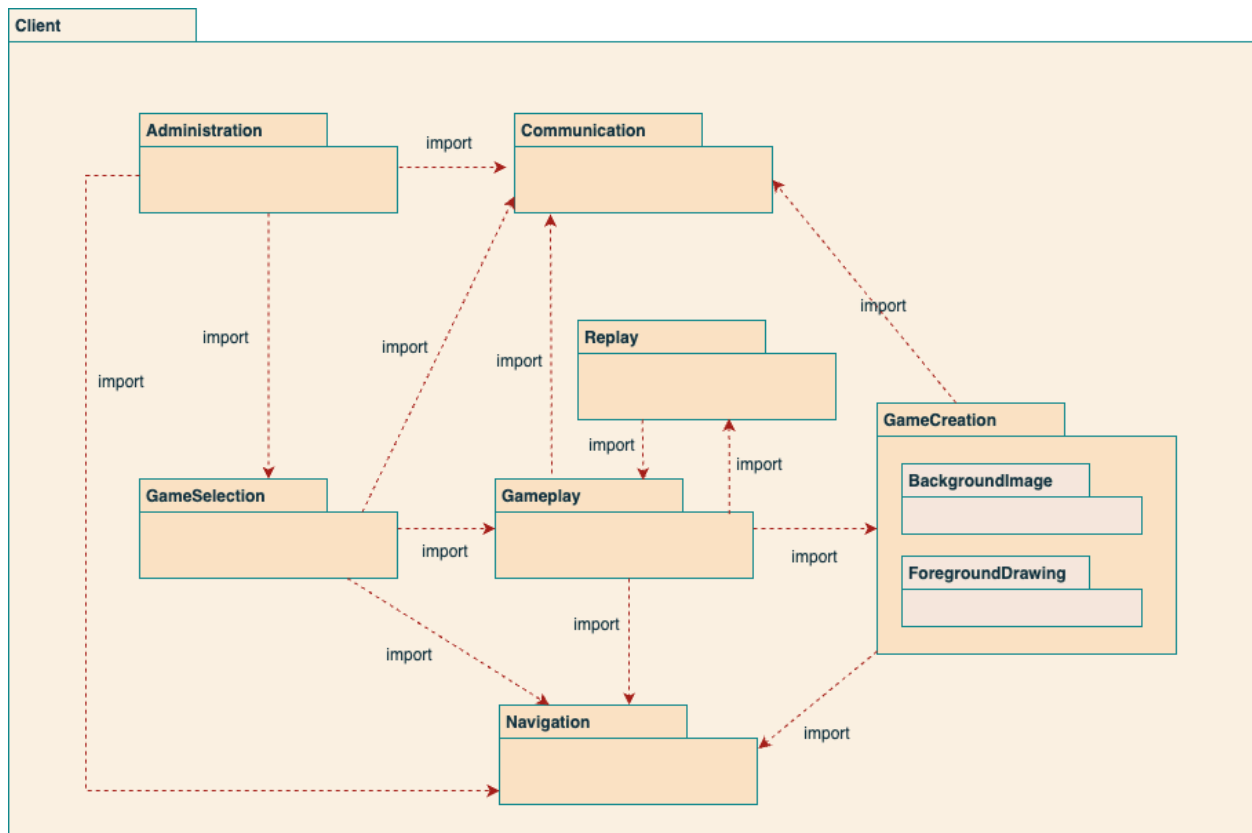
## 4. Vue logique

### Client

Ce paquetage contient l’affichage et la logique de l’application du côté client. Il est divisé en sous-paquetages qui se chargent des différentes fonctionnalités principales du site web. Ces sous-paquetages sont détaillés dans les sections qui suivent.

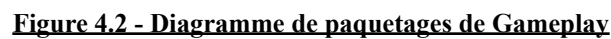
*Les classes sont volontairement omises pour alléger le diagramme. Elles sont présentées dans les diagrammes des différents sous-paquetages.*

Le client lourd et le client léger ont des architectures similaires, à l’exception que le paquetage GameCreation est présent uniquement dans le client lourd et pas dans le client léger.



**Figure 4.1 - Diagramme de paquetages de Client**

Ce paquetage se charge de la vue et la logique de jeu, aussi bien en mode classique qu'en mode temps limité. Il montre à l'utilisateur les différents éléments de rétroaction visuelle et sonore et il communique avec le serveur pour entre autres valider les clics et recevoir les événements liés au jeu. La communication avec le serveur se fait à travers les fonctions de `SocketClientService`, une classe externe à ce paquetage.





Ce paquetage se charge d'enregistrer les actions qui ont lieu durant la partie et de les rejouer pendant la reprise vidéo. Il permet de changer la vitesse d'exécution des actions, ainsi que de les mettre en pause, de reprendre leur exécution, et de recommencer la reprise vidéo depuis le début.

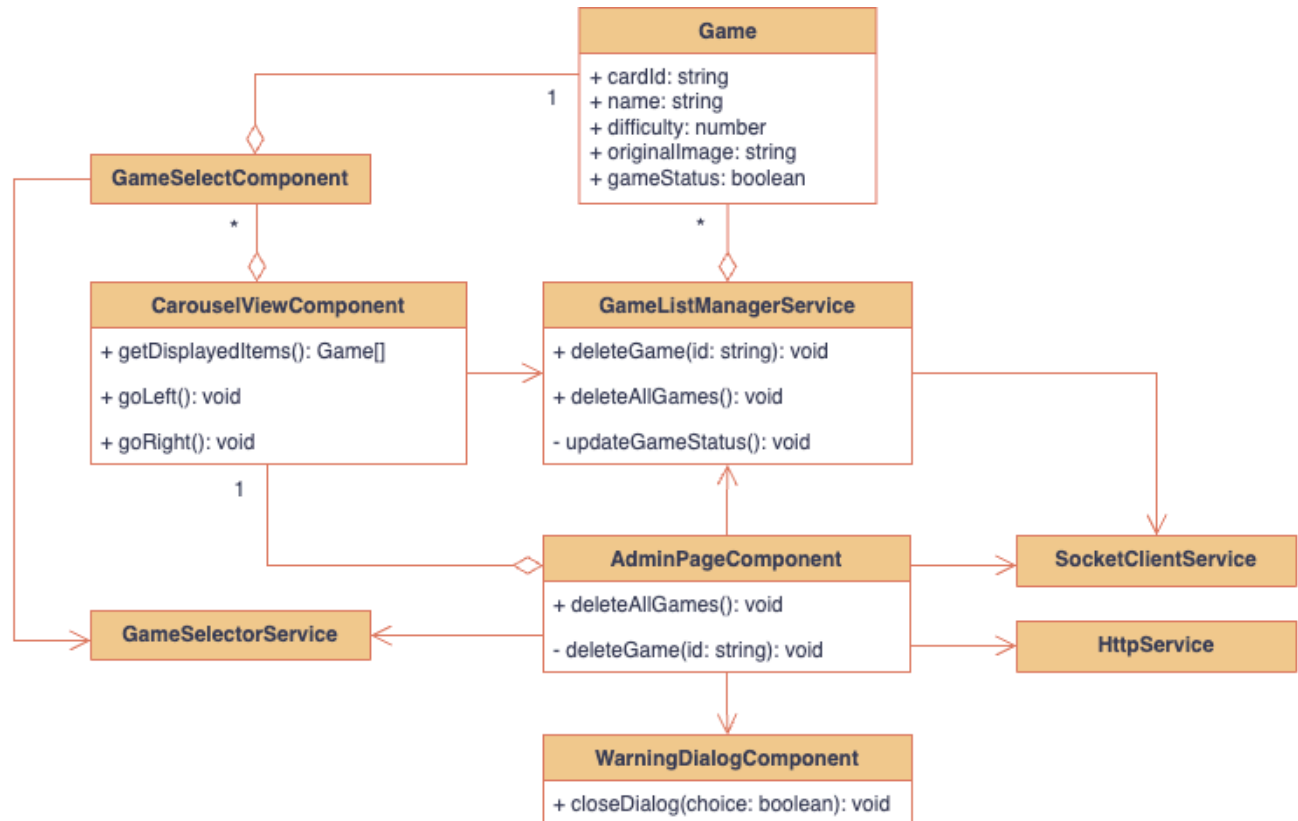


### Administration

Ce paquetage se charge de la vue d'administration du site, ce qui inclut :

- L'affichage des fiches de jeu disponibles
- La suppression des fiches de jeu

La création de jeu est couverte dans le paquetage GameCreation.

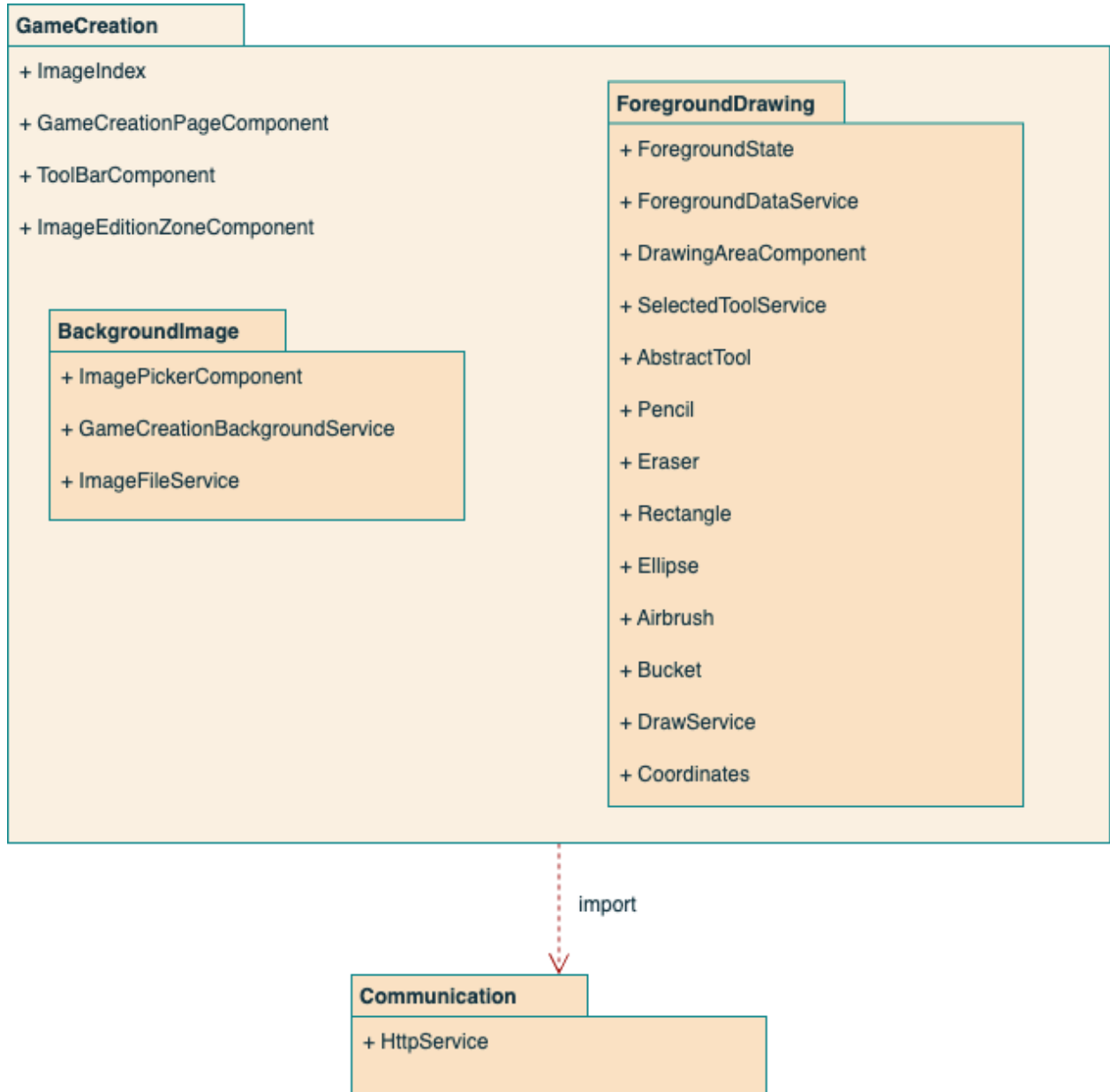


**Figure 4.5 - Diagramme de classes de Administration**

### GameCreation

Ce paquetage se charge de la fonctionnalité de création de jeu du côté client, c'est-à-dire le téléversement d'images d'arrière-plan, le dessin sur l'avant-plan, ainsi que l'envoi et la réception des informations du jeu à créer avec le serveur.

Ce paquetage est présent uniquement dans le client lourd. Il est absent du client léger.



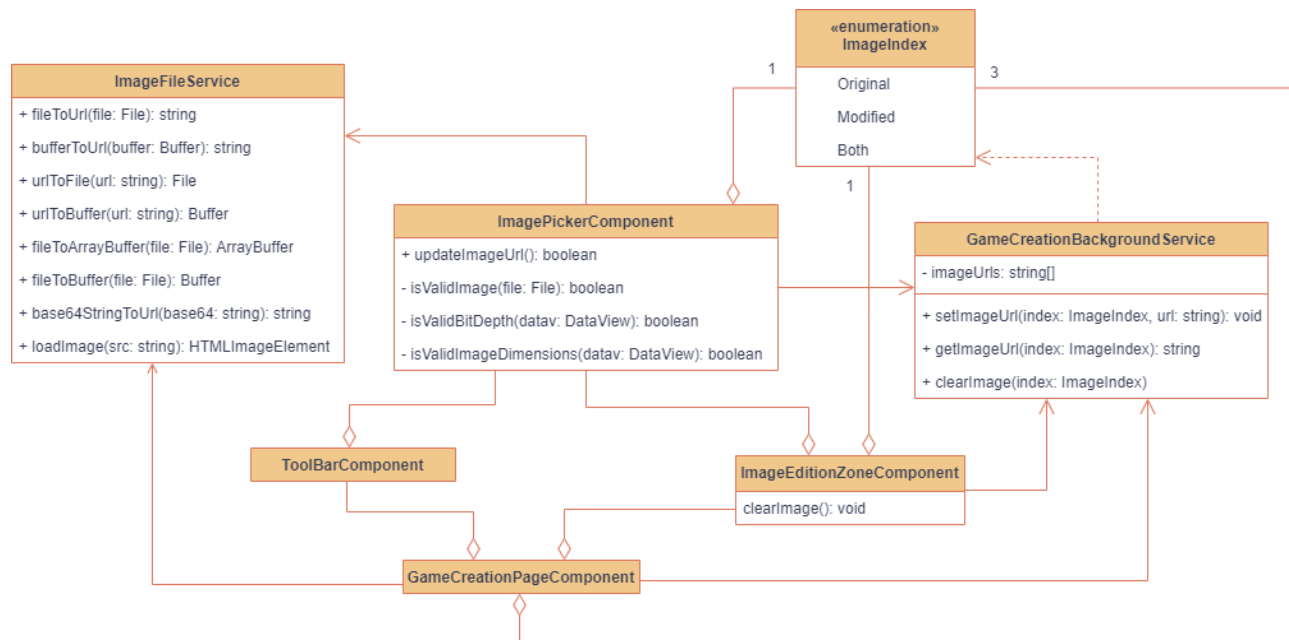
**Figure 4.6 - Diagramme de paquets de GameCreation**

### BackgroundImage

Ce paquetage se charge de la fonctionnalité de téléversement d'images d'arrière-plan, ce qui inclut :

- le téléversement d'images
- la validation des contraintes sur l'image
- la réinitialisation de l'arrière-plan

Ce paquetage est présent uniquement dans le client lourd. Il est absent du client léger.



**Figure 4.7 - Diagramme de classes de BackgroundImage**



## ForegroundDrawing

Ce paquetage se charge de la fonctionnalité de dessin sur l'avant-plan, ce qui inclut :

- le dessin au crayon
- l'efface
- le tracé de rectangle
- le tracé d'ellipse
- le seau de peinture
- le dessin à l'aérosol
- la duplication d'un avant-plan
- l'intervention des avant-plans
- la réinitialisation d'un avant-plan
- annuler et refaire une action

Ce paquetage est présent uniquement dans le client lourd.

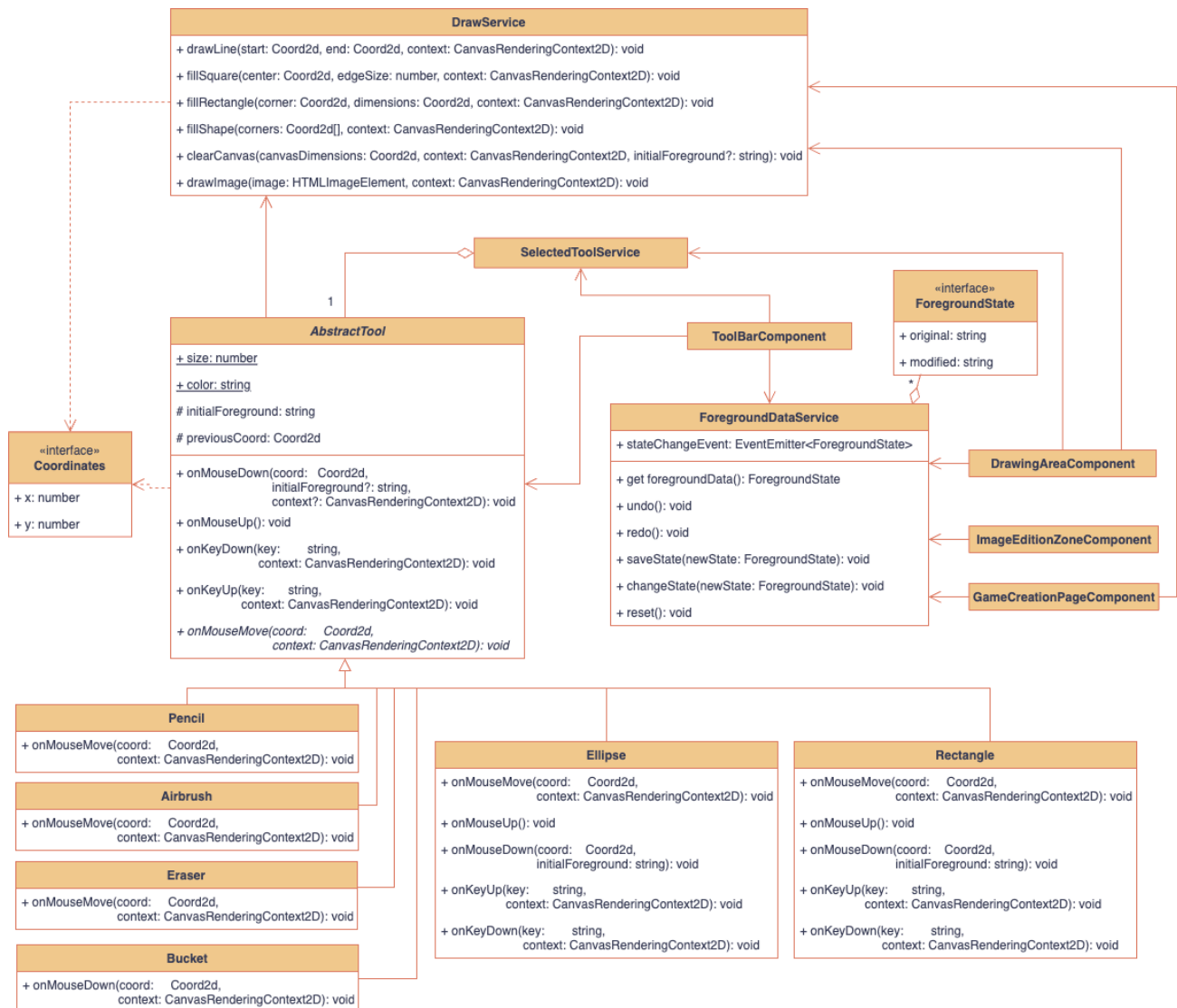
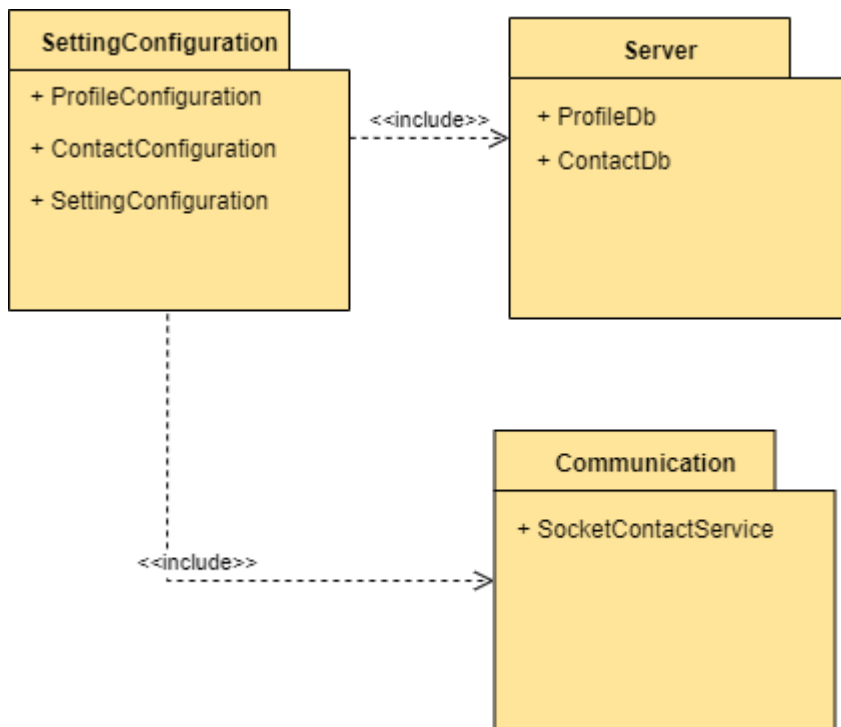


Figure 4.8 - Diagramme de classes de ForegroundDrawing

### SettingConfiguration

Ce paquetage se charge de la fonctionnalité des différents configurations dans le système, ce qui inclut :

- la configuration du profil
- la configuration des contacts
- la configuration des paramètres.

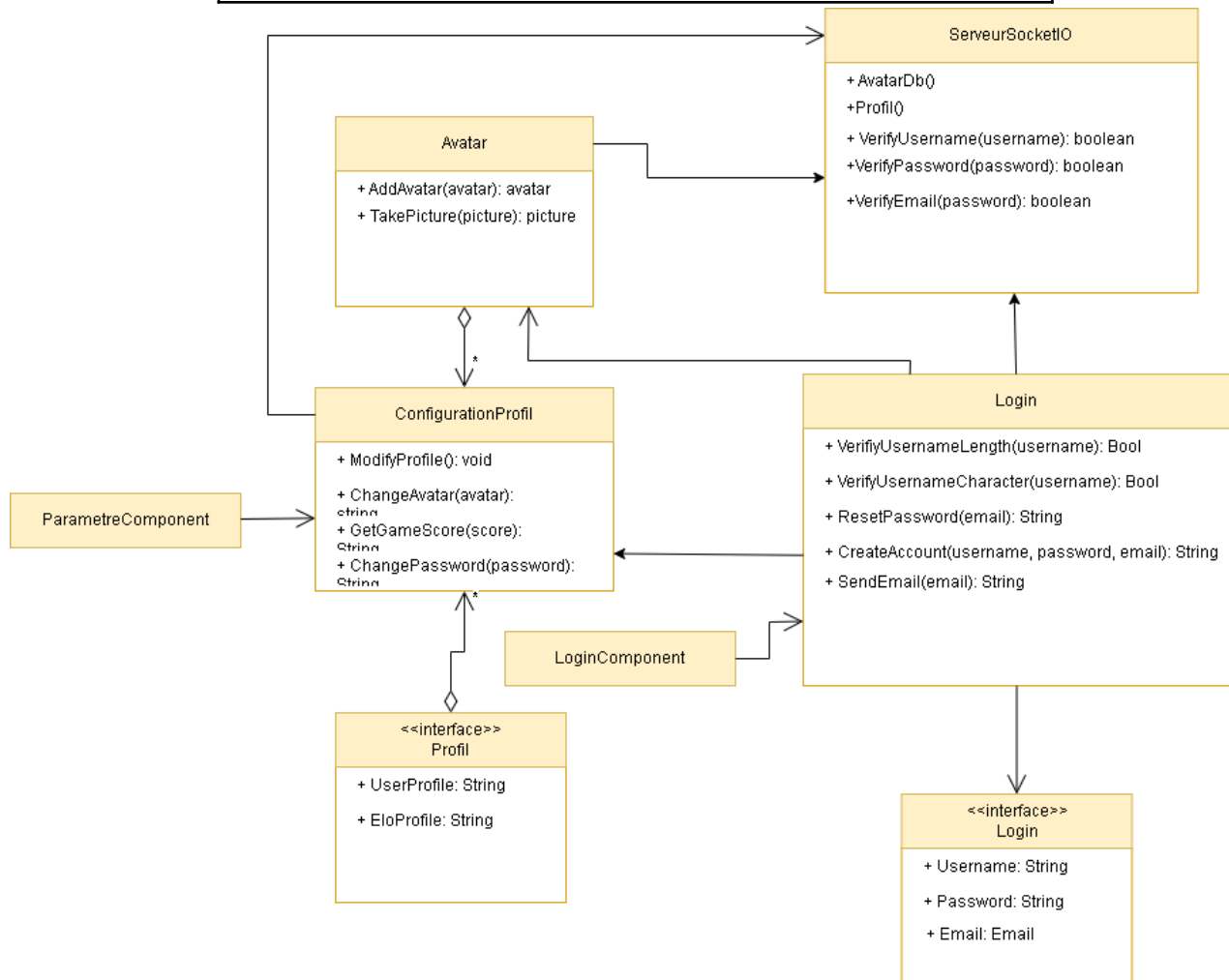


**Figure 4.9 - Diagramme de Paquetage pour les configurations**

### Configuration de Profil

Ce diagramme de classe se charge de la fonctionnalité des différents configurations en lien avec le profil, ce qui inclut :

- la configuration du profil
- l'utilisation des avatars
- le système de login

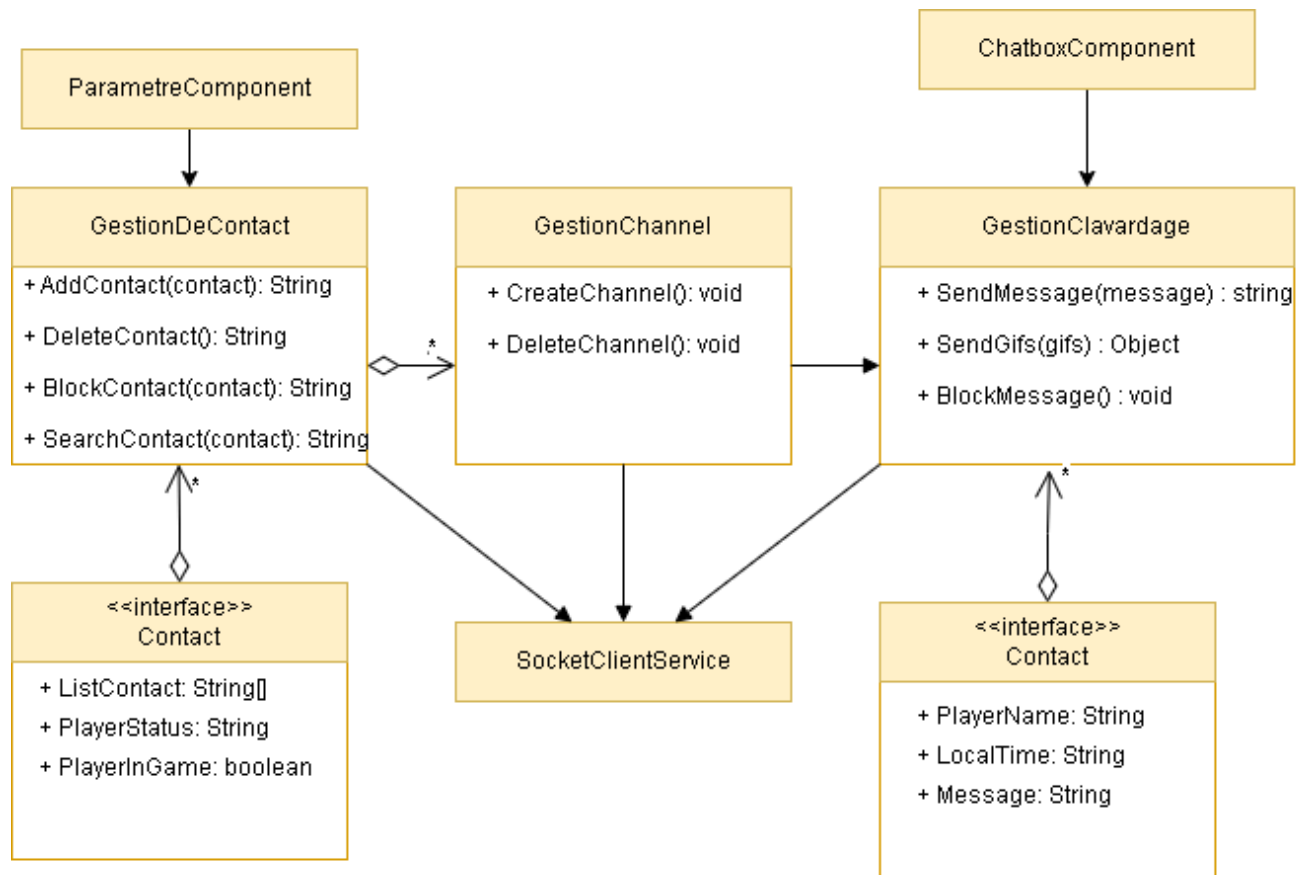


**Figure 4.10 - Diagramme de classe pour les configurations de Profil**

### Configuration de Profil

Ce diagramme de classe se charge de la fonctionnalité des différents configurations en lien avec les contacts, ce qui inclut :

- la configuration de la liste de contact
- la création de channel
- le système de clavardage

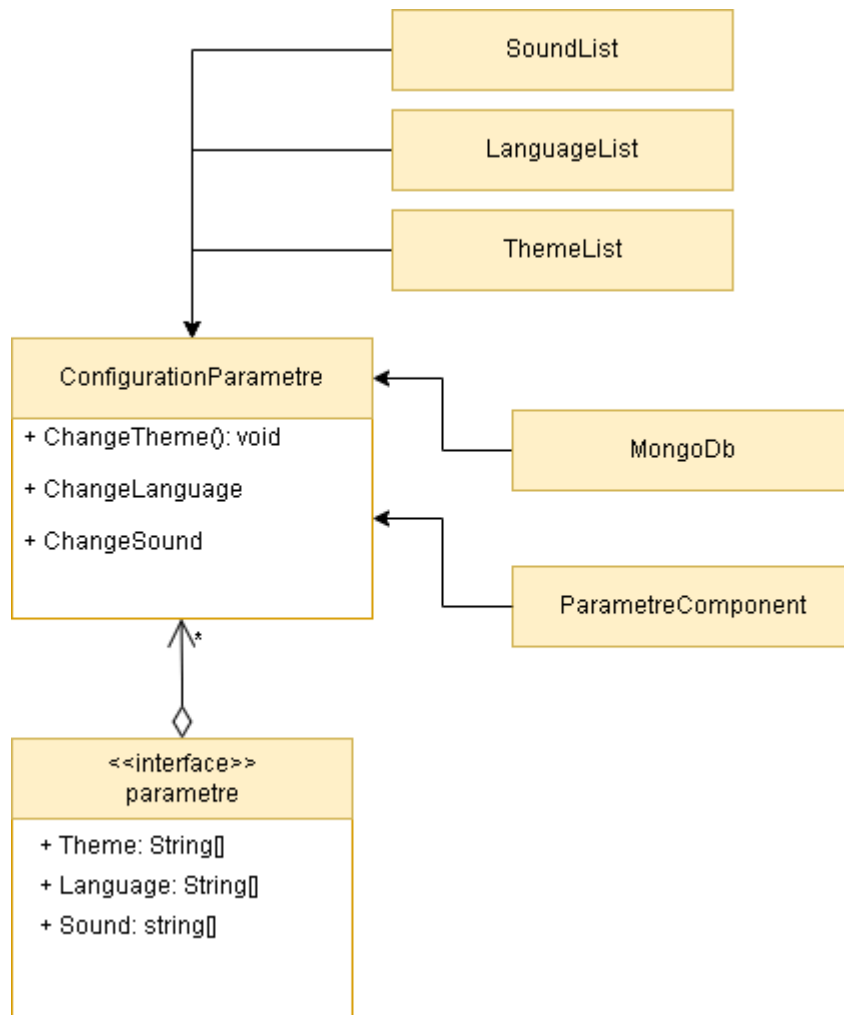


**Figure 4.11 - Diagramme de classe pour les configurations de contacts**

### Configuration de Profil

Ce diagramme de classe se charge de la fonctionnalité des différents configurations en lien avec les paramètres, ce qui inclut :

- la configuration de la langue
- la configuration du theme
- la configuration du song



**Figure 4.12 - Diagramme de classe pour les configurations de persistance**

## 5. Vue des processus

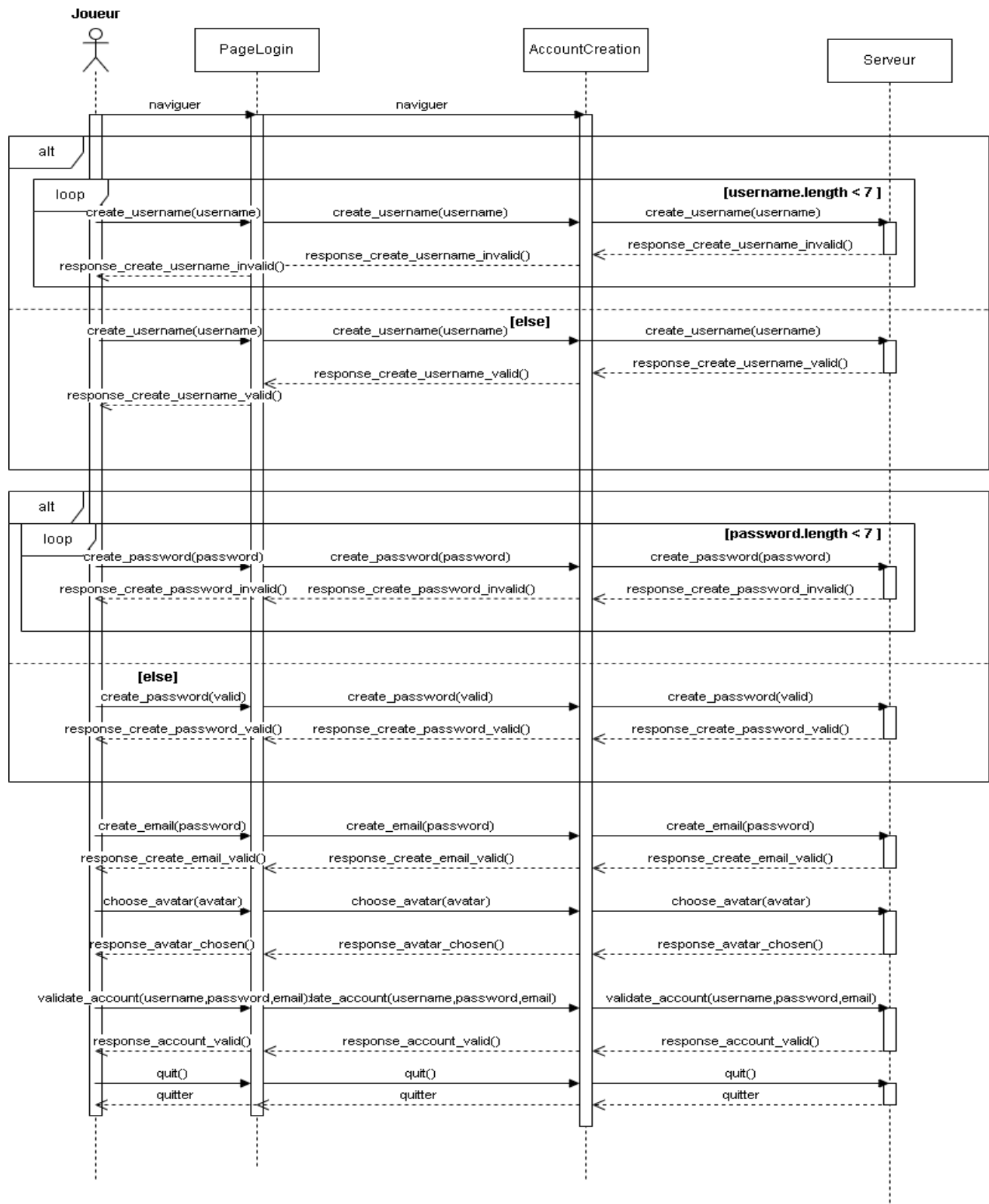
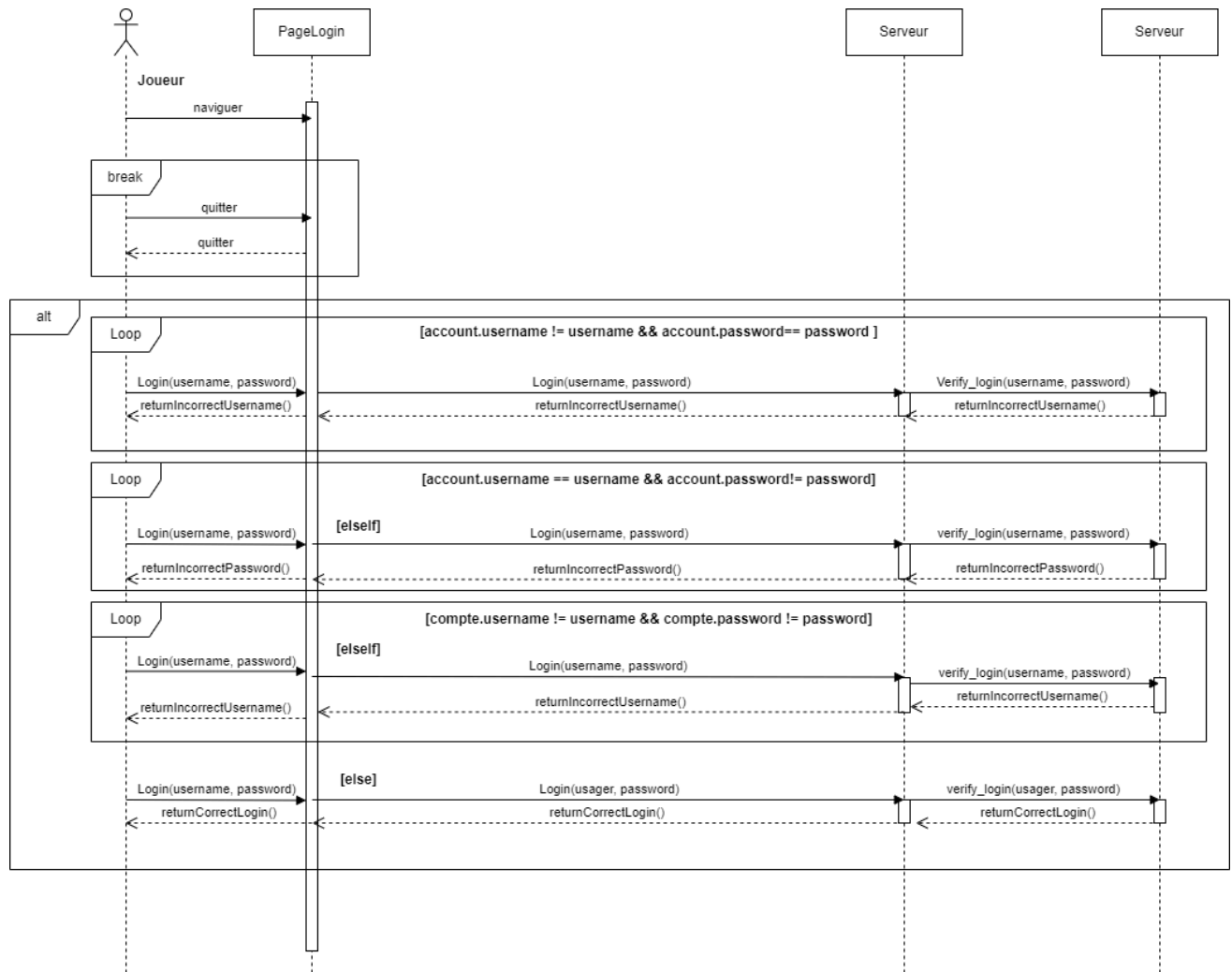
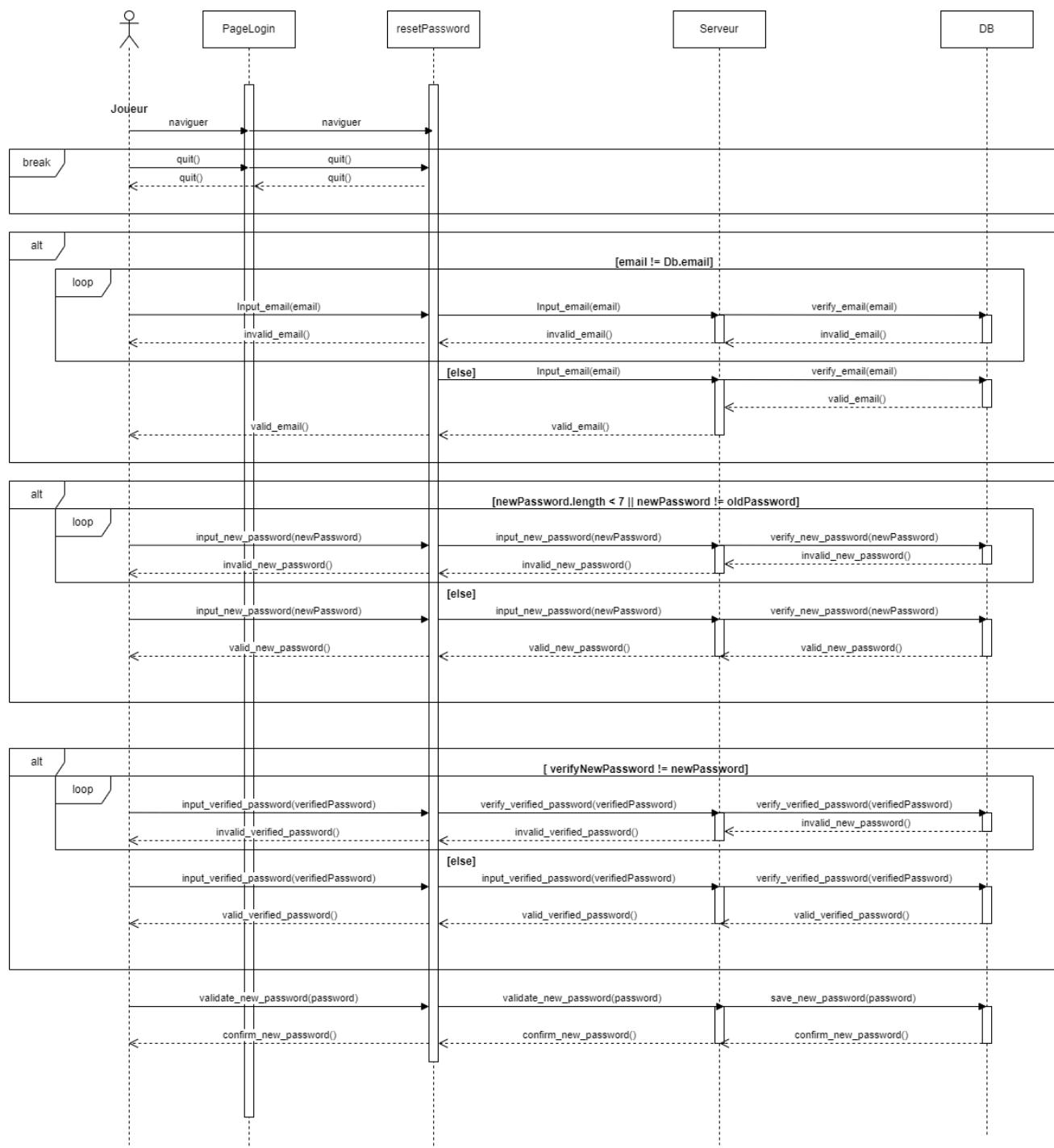


Figure 5.1: Vue de processus pour la page de création du profil.



*Figure 5.2: Vue de processus pour la page de login.*



**Figure 5.3: Vue de processus pour la page de login pour réinitialiser le mot de passe.**



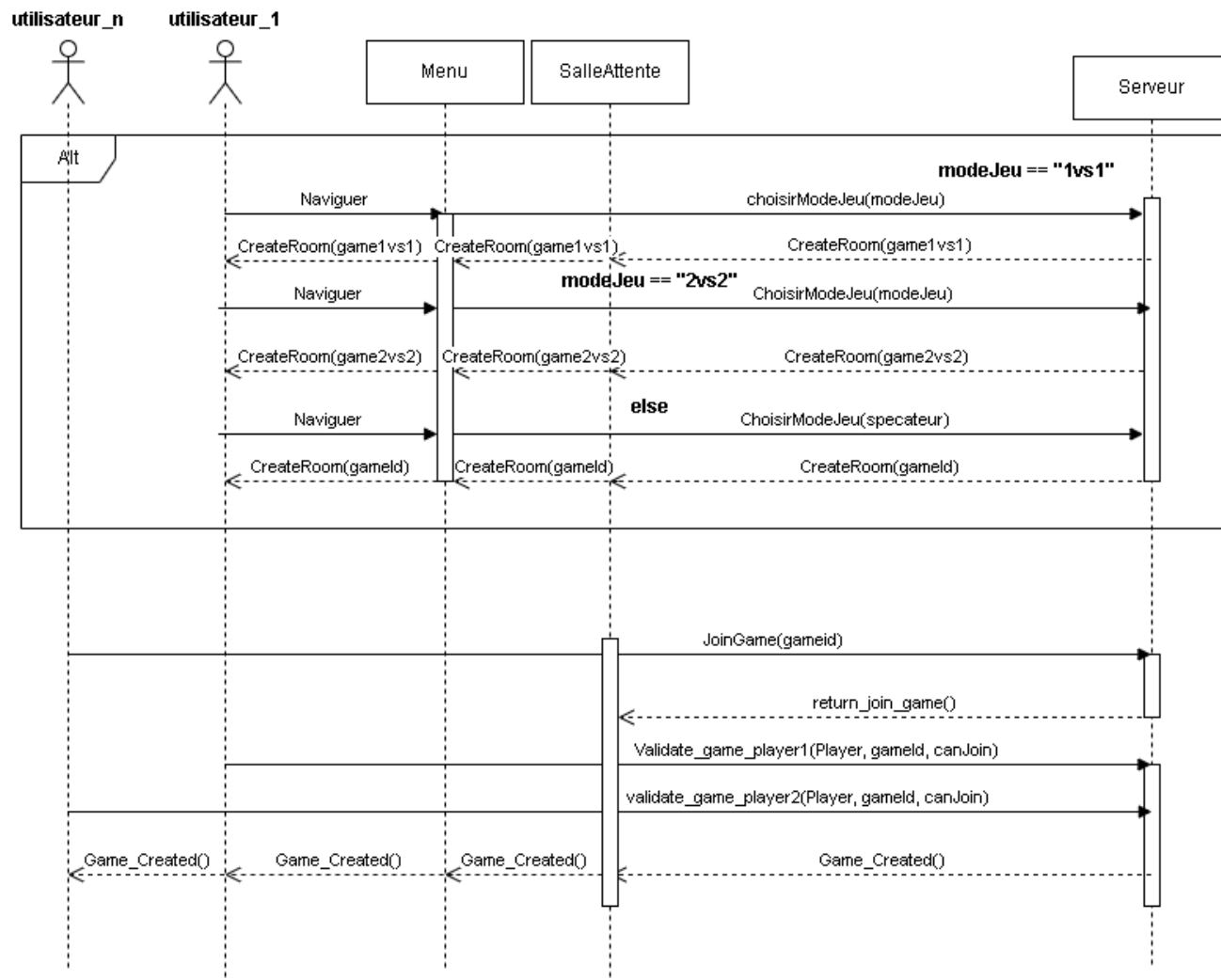


Figure 5.4: Vue de processus pour la page création d'une partie.

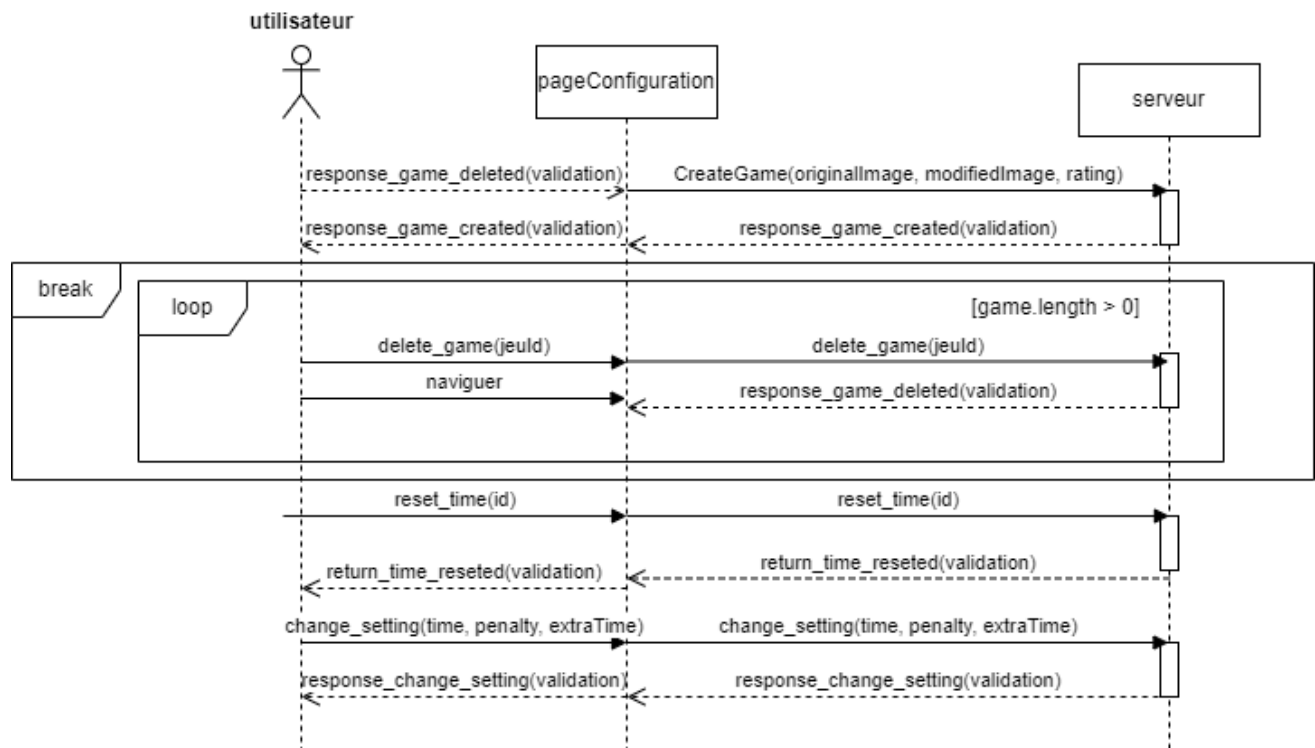


Figure 5.5: Vue de processus pour la page de configuration des jeux.

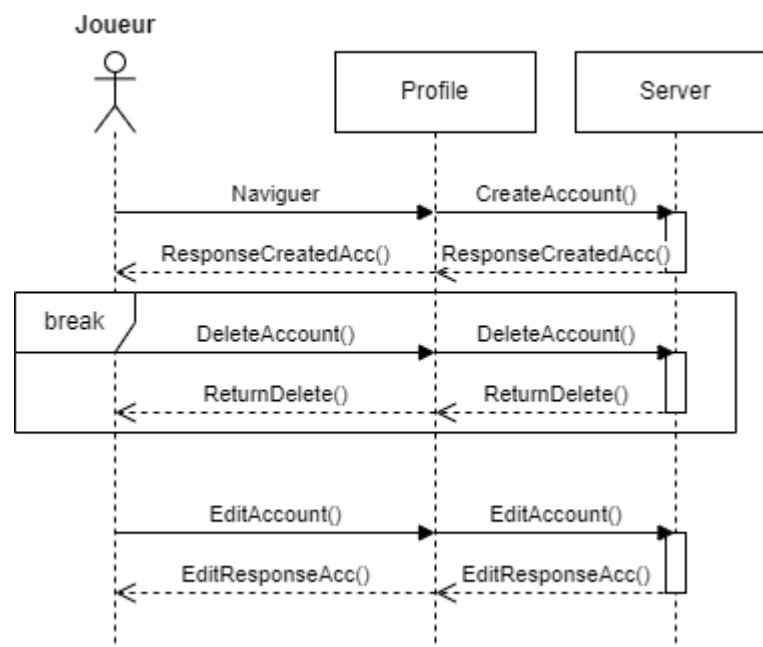


Figure 5.6: Vue de processus pour la page de profil.

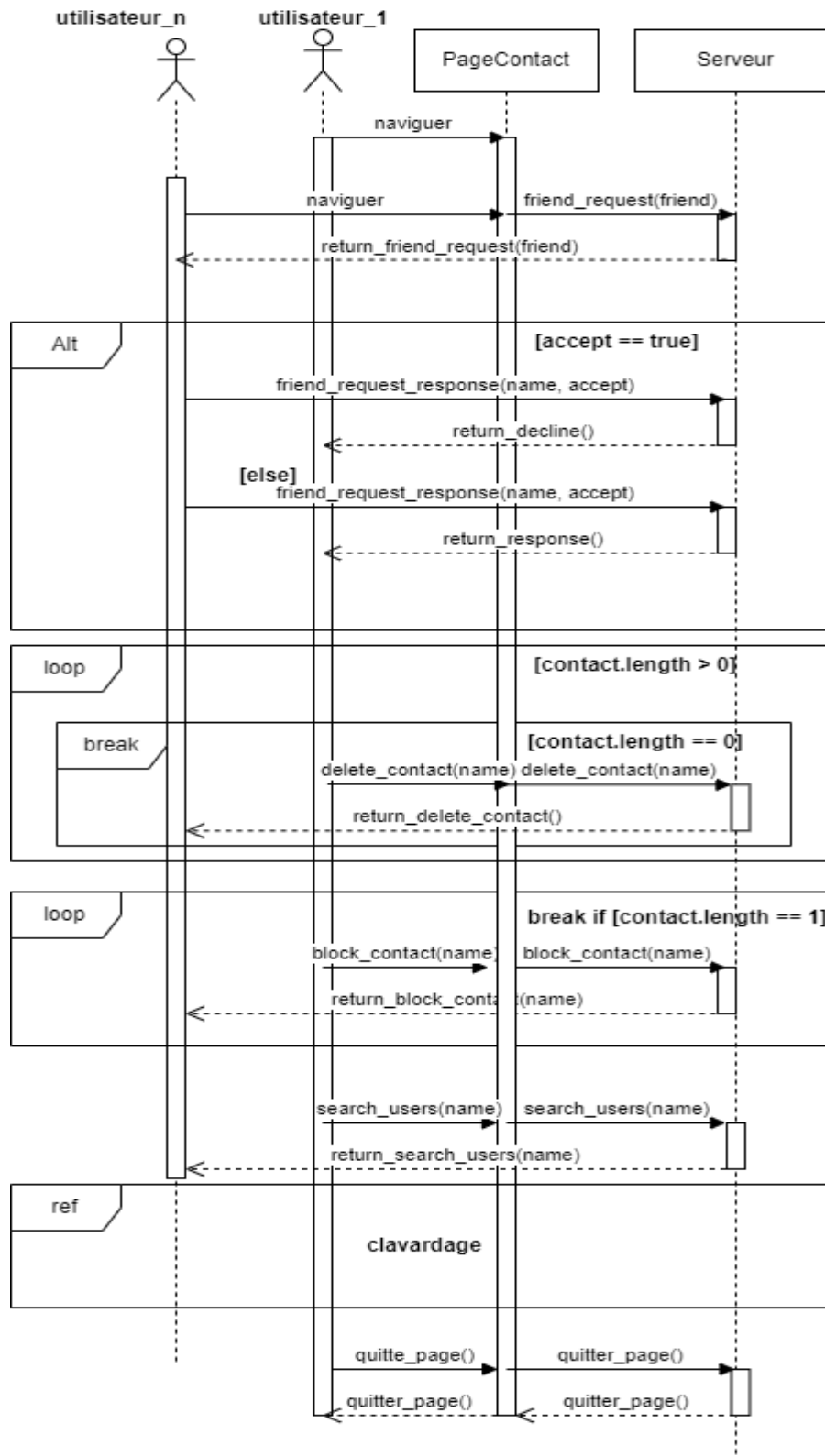


Figure 5.7: Vue de processus pour la page de contact.

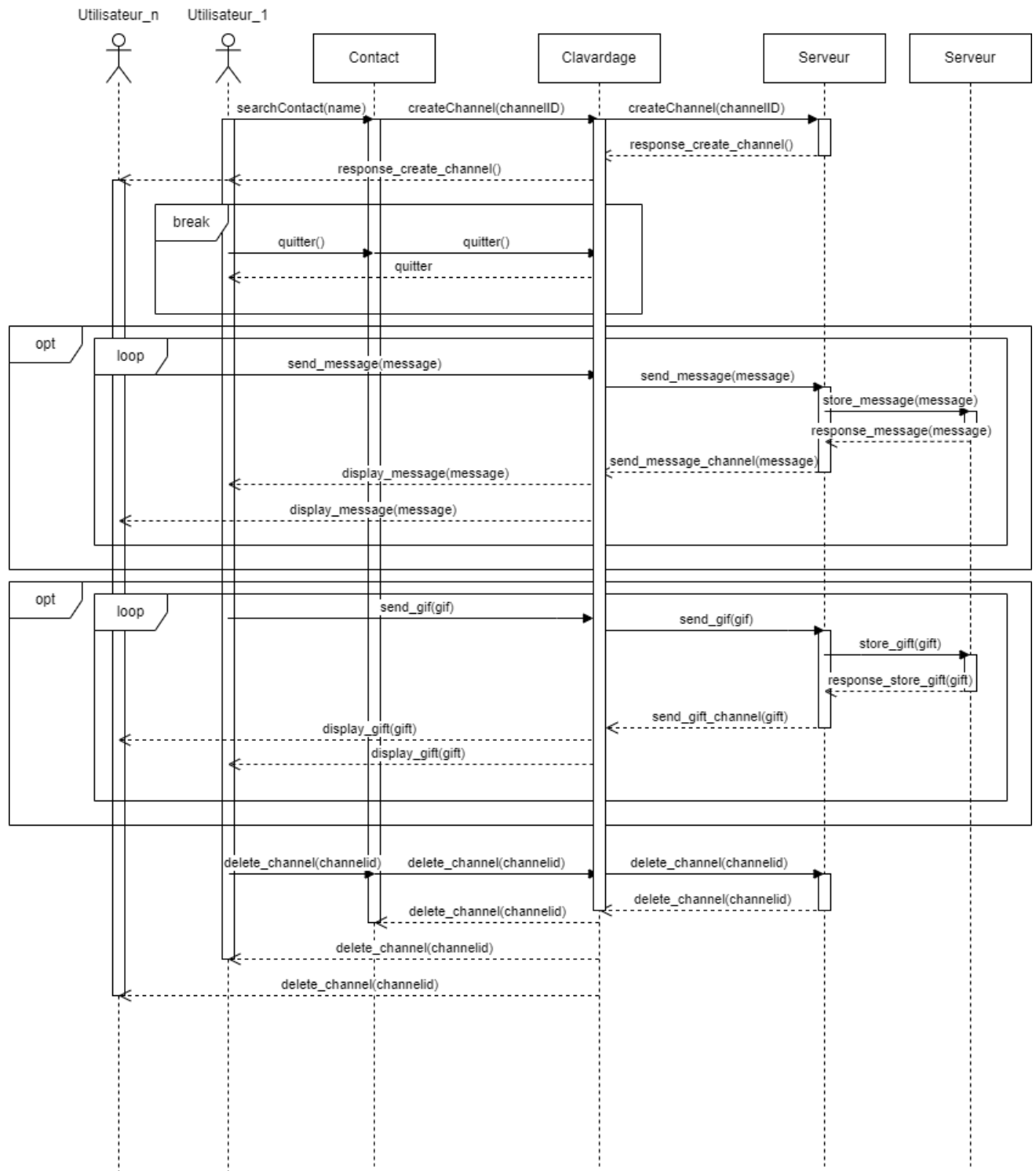


Figure 5.8: Vue de processus pour la page de message et channels.



## 6. Vue de déploiement

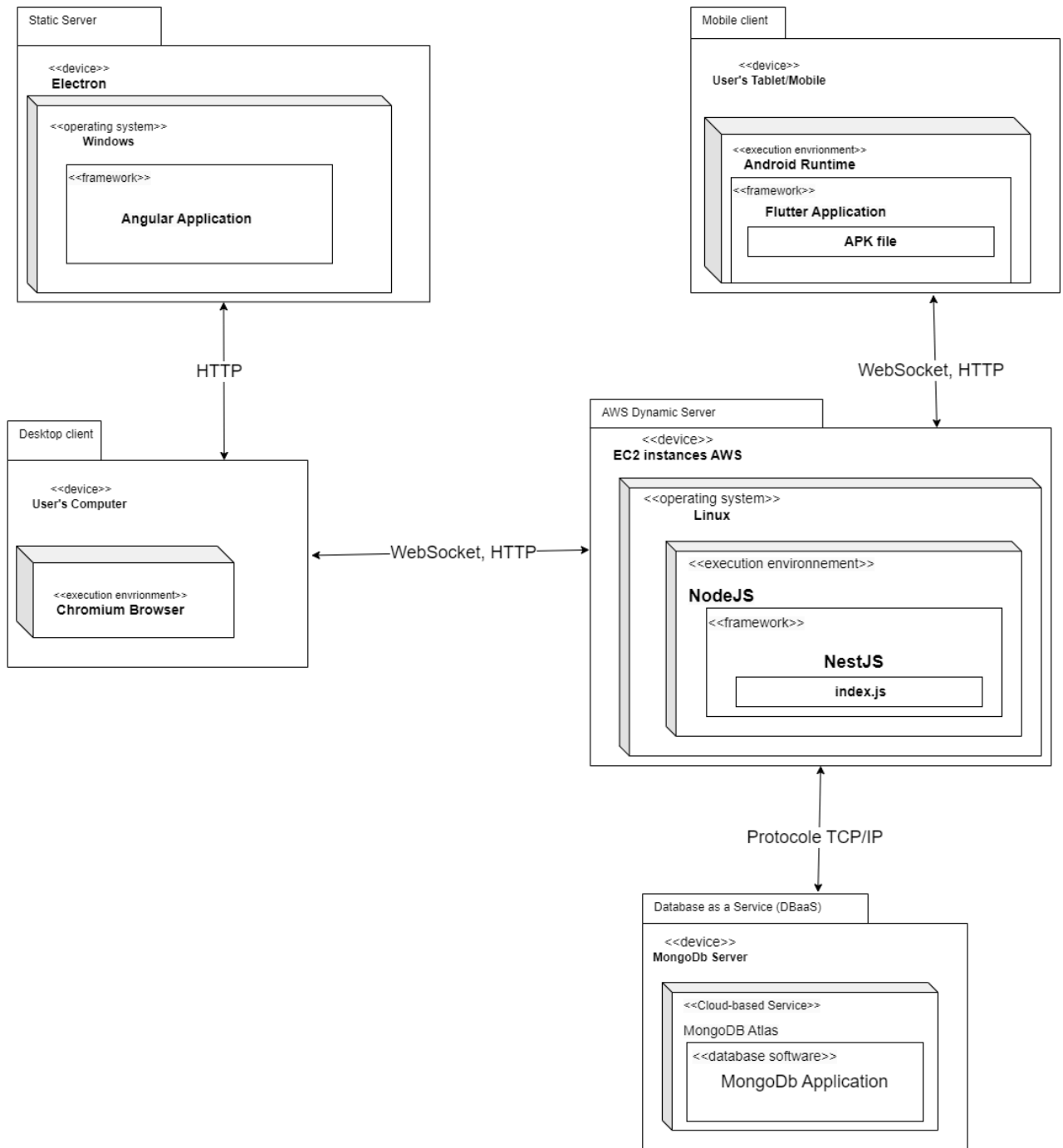


Figure 6.1: Vue du déploiement de notre système.

## 7. Taille et performance

Dans le développement de notre application, il est crucial de prendre en considération l'**espace disque disponible** sur les appareils cibles, tels que la tablette Samsung S6 Lite, qui dispose de 64 GB de capacité de stockage. Afin de garantir l'accessibilité de notre application sur une large gamme d'appareils, y compris ceux avec des capacités de stockage limitées, nous nous engageons à maintenir la taille de nos fichiers exécutables dans des limites raisonnables. Cela implique une optimisation rigoureuse du code pour éviter tout excès inutile qui pourrait encombrer l'espace de stockage de l'appareil. En privilégiant l'efficacité et la légèreté de nos applications, nous visons à maximiser la compatibilité et la performance, tout en minimisant l'impact sur l'espace disque des utilisateurs. Cette démarche s'inscrit dans une stratégie globale de développement respectueuse des ressources des appareils, assurant ainsi une meilleure expérience utilisateur pour tous.

Pour la **mémoire**, nous nous assurerons que la mémoire de notre application mobile ne dépasse pas 200 Mo et que celle de l'application de bureau ne dépasse pas 700 Mo. Cela permettra d'avoir un jeu accessible à une large gamme d'appareils surtout ceux avec des capacités de stockage limitées. Aussi, cela nous obligera à faire en sorte que nos applications aient une optimisation du code, un travail réduit en arrière-plan, que les ressources inutilisées seront libérées et aussi d'éviter les fuites de mémoire.

Pour ce qui est de l'**utilisation de ressources**, nous allons faire en sorte de développer une application avec des algorithmes optimisés et une faible complexité temporelle afin de minimiser l'utilisation du CPU. Nous veillerons d'ailleurs à ce que l'utilisation des ressources de nos serveurs ne dépassent pas 75% de leur capacité pour éviter des ralentissements ou des pannes de serveur.

Enfin pour la **vitesse d'exécution**, comme notre application de jeu est en ligne et a besoin de s'échanger des informations en temps réel que ce soit dans le mode de jeu temps limité 1v1 ou le clavardage, nous avons pour objectif d'avoir un temps de réponse en dessous de 1 seconde afin de garantir une expérience utilisateur fluide avec une latence minimale. Le temps moyen pour charger les images d'une fiche de jeu ne devrait pas dépasser 3 secondes.